

# **EXHIBIT 14**

From: Daniel Diez.  
 To: [ - ] Dan Bornstein.  
 Cc: [ - ] noser@google.com.  
 Bcc: [ - ]  
 Subject: Re: JavaDoc.

Sent: 5/18/2007 12:00 PM.

Dan

Thanks for the prompt answers.

With ref to security I copy below the mail. Specially important is the minimal functionality (2).

I also copy some design documentation below.

Best regards,  
 - Daniel

PS: I will send you the addresses for noser@google.com next Monday.

Begin forwarded message:

From: Daniel Diez <diezdani@gmail.com>  
 Date: May 9, 2007 8:02:49 PM GMT+02:00  
 To: danfuzz@google.com  
 Subject: Security questions for discussion

- Daniel

Begin forwarded message:

From: "Gruntz Dominik" <dominik.gruntz@fhnw.ch>  
 Date: May 9, 2007 7:10:53 PM GMT+02:00  
 To: "Daniel Diez" <diezdani@gmail.com>  
 Subject: Symphonie security questions

We give you a not exhaustive first list of questions:

(1) Normally as soon as the SecurityManager is active, the JVM picks up the text file "java.policy" to define the security rights of the current app.

Q: Do we need the flexibility offered by the java.policy file or can we setup a standard fixed security environment valid for every application? [i.e. do we need to be standard conform or not in this point]

(2) We suppose that only a part of the standard security functionality of Java 1.4.2 / 1.5 is required. Does Google have concrete ideas about

this set or should we make a proposal to be approved by Google?

A proposal to discuss:

Minimal core functionality: AccessControl, doPrivileged, DomainProtection, Activation of the SecurityManager via application.

SSL layer: Java https functionality (Server and client).

Certificate verification only on the server side. Are client X.509 certificates necessary?

Authentication: Part of SASL is already implemented (in our o

pinion is the full JAAS not necessary).

Authorization: Minimal solution via SecurityManager (full via JAAS)

Cryptography: Native methods are employed for SHA-1 and MD5 (.../core/android/security). We could take parts from the BouncyCastle distribution to do the same in standard Java (or harmony code).

(3) In our opinion it is not possible to activate the SecurityManager before the application starts on the device: is that correct or did we miss something?

(4) In our opinion doPrivileged is not implemented and we do not see a possible interface to native methods. Are we correct?

In case that doPrivileged is not supported by the JVM, what are the plans by Google? Are we expected to add this functionality to the kernel?

(5) How can we use the dalvik JVM in isolation in order to test its functionality? The idea is to install it in eclipse and so test it with Harmony (or to start the JVM from within a console).

(6) Can you give us some pointer about the functionality of the native methods used by the AccessController in dalvik VM? We found something (TreeWalker) but it is still very dark. We need more information on this point.

Status Report Mai 11 ¶

We looked again into security today and learned aspects which will be documented on the wiki soon.

However, we got the impression, that it would be the best way to port the harmony implementation onto Android. According to the current state of our analysis the harmony implementation depends on only one native method which would have to be provided by google (or which would have to be implemented by us). This method returns the active ProtectionDomains. We will further investigate next week what exactly this method returns.

The analysis of the harmony code showed, that the doPrivileged methods are implemented in java code (in cooperation with the AccessController class), so we propose to go that way as well as a first approach.

Regarding question (6): We found the Java class VMStackWalker (dalvik/libcore/luni/src/main/java/java/lang/VMStackWalker.java) with the following 3 native methods:

- getClassContext();
- getCallingClass();
- getCallingClassLoader();

Probably, the required hook which is used by harmony could be implemented based on the VMStackWalker, but we would need further information on this class.

We found a C implementation of these methods in the file: extlibs/jamvm-1.4.1/src/natives.c But this is a code file from the JamVM project (<http://jamvm.sourceforge.net>). => GPL / Licence Problems !!!

When we try to call e.g. the method getClassContext(), the virtual machine returns follow error:  
java.lang.UnsatisfiedLinkError: getClassContext is there an implementation of the methods for the dalvik vm

available?

Once we agreed that the way with the harmony port is ok we still need an answer from Google which features should be eventually provided on the device, i.e. we could imagine that a set of standard policies could be hard coded. Probably other restrictions of the security features are desired.

On May 18, 2007, at 6:34 PM, Dan Bornstein wrote:

On 5/17/07, Daniel Diez <diezdani@gmail.com> wrote:

How are you doing?

Pretty well, thanks! I see you sent several emails; I'll try to answer all of them reasonably quickly.

- Besides Harmony it is allow 'copy' the JavaDocs from Classpath?

Absolutely not. The Classpath code -- including the comments -- is covered by an incompatible license.

- At the moment there is no generation of javadocs for the java corelibs. Should be include this generation in a Makefile?

I hadn't noticed that (I guess I usually just browse the sources directly). I will make sure that that gets added to our build.

Any news about the Security minimal implementation?

I thought you were going to send a proposal to kick it off, based on our discussions. My apologies if I am mis-remembering. Let me know.

-dan

Daniel Diez  
diezdani@gmail.com  
fix +41 44 586 4654  
mobile +41 78 825 9155  
skype daniel\_diez  
Zurich - Switzerland